REMARKS

Claims 1-20 are pending. Claims 1, 2, 4-6, 8, 9, and 14-16 have been amended. No new matter has been added.

Claims 1-20 are rejected under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter of the invention. The Applicant is advised of the obligation under 37 C.F.R. 1.56 to point out the inventor and invention dates of each claim that was not commonly owned at the time a later invention was made. Claims 1-20 are rejected under 35 U.S.C. 103(a) as being unpatentable over Adams, et al. (Publication No. 2003/0070157), in view of Bowman-Amuah (Patent No. 6,256,773).

The undersigned respectfully requests consideration of the arguments presented herein and withdrawal of the present rejections.

Rejection of Claims 1-20 under 35 U.S.C. 112, second paragraph,

The undersigned has removed references to the word "approximate" and "approximating" throughout the claims in order to overcome this rejection.

Rejection of Claims 1-20 under 35 U.S.C. 103(a) as being unpatentable over Adams, et al. (Publication No. 2003/0070157), in view of Bowman-Amuah (Patent No. 6,256,773).

The Office relies primarily on the Adams reference in rejecting claims 1-20. Importantly, Adams is directed to calculating an estimated effort to maintain an existing software system. While one factor used to calculate this estimated maintenance effort is productivity level and, one identified technique for calculating the productivity level uses a capability maturity model, Adams clearly does not disclose the claimed invention. Again, estimating a maintenance effort is not the same as determining maturity in accordance with a particular maturity model.

Starting with the preambles of the independent claims, the Office cites to the following paragraphs of Adams as teaching determining maturity in accordance with a particular capability maturity model:

[0031] The present invention provides a system and method for estimating effort and cost to maintain a software application, group of applications or an aggregate system of applications (each of which is referred to herein as a "system"). A system size and productivity level are determined. The productivity level preferably takes into consideration the maintenance tasks to be performed as well as personnel attributes, such as capability and experience pertaining to the task. The effort equals the product of an effort multiplier and the system size divided by the productivity level. The effort multiplier preferably takes into account maintenance complexities that may result in added effort and cost. The cost is determined by applying prevailing rates and fees to the calculated effort. As the maintained system is developed and enhanced, and as portions of the system are

retired, the system size and productivity level are reassessed and the effort and cost to maintain the system as modified are re-computed.

[0057] A second technique for calculating a maintenance productivity level, which is preferred, involves calculating a net maintenance productivity ratio and applying a COCOMO II-based effort adjustment factor. The original COCOMO constructive cost model, first presented by Dr.

Barry Boehm in "Software Engineering Economics," Prentice Hall, Englewood Cliffs, N.J., 1981, provided a structured methodology for estimating cost, effort and scheduling in planning new software development activities. COCOMO II, a revised model, emerged to reflect changes in professional software development practice since the original model.

[0058] To calculate the productivity level, average productivity ratios (FP/FTE) are applied to the maintenance tasks comprising the maintenance effort. **FIG. 10** provides a table of such productivity ratios based on Jones, T. Capers, "Estimating Software Costs," McGraw Hill, New York, N.Y., 1998, Table 27.3, p.600. The first column identifies common maintenance tasks. The second column provides, as a productivity ratio, the number of function points one FTE (e.g., a maintenance programmer at 152 hours per month) can handle for the task.

[0059] The average productivity ratios are then weighted, according to the estimated percentage each task will comprise of the total maintenance effort, as shown in the third column of **FIG. 10**. Then, weighted averages (column **4**) are calculated by dividing the percentage (column **3**) by the average productivity ratio (column **2**), as shown in **FIG. 10**. Next, the weighted averages are summed. The net maintenance productivity ratio equals the inverse of the sum of the weighted averages.

[0060] Finally, the net maintenance productivity ratio is divided by a COCOMO II-based effort adjustment factor, resulting in the maintenance productivity level. The effort adjustment factor (EAF) is determined based on COCOMO II personnel attribute cost drivers, as shown in **FIG. 8**. The effort adjustment factor equals the product of the applicable effort ratings for the personnel cost drivers. Effort multipliers may be determined via interpolation or extrapolation for percentiles not provided in the table.

[0061] A third technique for calculating a maintenance productivity level involves correlating productivity levels with the size of the maintenance task and the maturity level of an organization. The Capability Maturity Model for Software (CMM), developed by the Carnegie Mellon Software Engineering Institute, provides a preferred model for judging the maturity of the software process. Each maturity level of the CMM corresponds to an evolutionary plateau toward achieving a mature software process. Referring to **FIG. 3**, by determining the scope of maintenance activities (left column) and the maturity level of an organization (columns **2**, **3** & **4**), a productivity level, which is based on historical evidence, can be determined.

Importantly, and as was previously stated, these paragraphs only refer to capability in the context of determining productivity level and the only paragraph that refers to a capability maturity model is paragraph [0061]. The Office suggests that COCOMO is a maturity model, when it is not. COCOMO is a constructive cost model used to estimate effort, cost and schedule for software projects. Accordingly, Adams simply does not teach any of the limitations of the claims that refer to or relate back to at least one maturity model. In fact, the undersigned submits that Adams is not directed to an analogous art. The mere mention of CMM does not mean that the reference is in an analogous art. Further, since the claims are specifically directed to

comparing the requirements of the maturity model to various generalized and specific work products of the company and Adams is not directed to using a maturity model (with the exception of the limited use in paragraph [0061]), again, the undersigned submits that the majority of the claimed limitations are not disclosed in Adams.

In combination with Adams, the Office cites Bowman-Amuah as teaching various implementation components of the claims, e.g., user interface, computer processor, use interface, keyboard, network connection and a port. These teachings do not cure the deficiencies of Adams with respect to the missing claim limitations. Accordingly, the undersigned submits that the combination of Adams and Bowman-Amuah does not render claims 1-20 unpatentable.

CONCLUSION

The undersigned representative respectfully submits that this application is in condition for allowance, and such disposition is earnestly solicited. If the Examiner believes that the prosecution might be advanced by discussing the application with the undersigned representative, in person or over the telephone, we welcome the opportunity to do so. In addition, if any additional fees are required in connection with the filing of this response, the Commissioner is hereby authorized to charge the same to Deposit Account 50-4402.

Respectfully submitted,

Date: August 10, 2009
By: */Dawn-Marie Bey - 44.442/*

Dawn-Marie Bey
Registration No. 44,442
King & Spalding LLP
1700 Pennsylvania Avenue
Suite 200
Washington DC 20006
(202) 626-8978